The 6th International Conference on Ambient Systems, Networks and Technologies
(ANT 2015)

A contactless identification system based on hand shape features

Ana M. Bernardos , Jose M. Sánchez, Javier I. Portillo, Juan A. Besada, José R. Casar

Abstract

This paper aims at studying the viability of setting up a contactless identification system based on hand features, with the objective of integrating this functionality as part of different services for smart spaces. The final identification solution will rely on a commercial 3D sensor (i.e. Leap Motion) for palm feature capture. To evaluate the significance of different hand features and the performance of different classification algorithms, 21 users have contributed to build a testing dataset. For each user, the morphology of each of his/her hands is gathered from 52 features, which include bones length and width, palm characteristics and relative distance relationships among fingers, palm center and wrist. In order to get consistent samples and guarantee the best performance for the device, the data collection system includes sweet spot control; this functionality guides the users to place the hand in the best position and orientation with respect to the device. The selected classification strategies - nearest neighbor, supported vector machine, multilayer perceptron, logistic regression and tree algorithms - have been evaluated through available Weka implementations. We have found that relative distances sketching the hand pose are more significant than pure morphological features. On this feature set, the highest correct classified instances (CCI) rate (>96%) is reached through the multilayer perceptron algorithm, although all the evaluated classifiers provide a CCI rate above 90%. Results also show how these algorithms perform when the number of users in the database change and their sensitivity to the number of training samples. Among the considered algorithms, there are different alternatives that are accurate enough for non-critical, immediate response applications.

Keywords: Biometry, hand-shape based identification, classification, smart spaces.

1. Introduction

The increasing availability of low-cost technology that makes possible to interpret in-air gestures, such as compact devices including depth sensors, facilitate the creation of different concepts of interaction. Although these devices are often designed to interface with laptops or applications, they have a great potential as tools to be integrated in our daily life environments for non-critical applications. In particular, algorithms for gesture recognition are usually categorized as user-independent or user-dependent, depending on their need for previous training. User-dependent ones usually provide a better accuracy[1], even if the additional training effort may reduce the system usability perception. Our previous works in this area have led us to search for a confortable identification mechanism, compatible with infrastructure devices enabling gesture recognition and delivering user identification without the need to equip the user with any extra device (e.g. wearable accelerometers).

The objective of this work is then to explore the possibilities of delivering a functional identification method based on contactless hand shape analysis on a Leap Motion sensor. Although this sensor is being widely used for interaction, to the best of our knowledge there are not empirical studies in the identification issue. The paper is oriented to compare the performance of different classification algorithms regarding accuracy, training needs and scalability. To introduce this comparison, Section 2 provides a review of previous research on shape hand-based identification. Section 3 defines the identification strategy itself, starting by the hand features to be used. Within this Section, it is also described the classification algorithms and the sweet pose tool used to gather the testing dataset. On this dataset, Section 4 presents a performance comparison that analyzes the information within the selected hand features; it also evaluates specific implementations of the algorithms, taking into consideration their accuracy, time to build the model, need for training and scalability. Finally, Section 5 discusses on results and further work.

2. Related work

Shape-based hand recognition is one of the first live biometrics-based recognition systems. The first systematic system to capture hand and finger images is dated in 1858[2]. In the mid 1960's, Robert Miller invented a mechanical hand geometry identification device[3]. The first commercial device (Identimate)[4] used mechanically scanned photocells to measure the finger length, the endpoint contours and the skin translucency. This device was in use from 1970's to 1987. In 1986, Recognition System presented the ID3D HandKey[4], the first device using low-cost digital imaging sensors. Currently, the increasing number of commercial systems and patents demonstrates the effectiveness of this biometric approach, with many approaches proposed and evaluated (e.g. Kong et al.[5] and Duta[6] provides complete surveys).

Many hand-based biometric schemes work obtaining geometric measures of the hand and then extracting a set of features from these geometric measures. The main hand recognition approaches are based on hand geometry, hand contour and palm print. The hand geometry systems use only hand geometric features, for instance, finger lengths, finger widths, palm areas, measure ratios, etc. These methods reduce the information given in a hand sample to a N-dimensional vector that is used to implement a matching algorithm based in a metric distance. Other alternative schemes are proposed in literature applying different probabilistic and machine learning techniques, like k-Nearest Neighbours[7], Gaussian Mixture Models[8], or Support Vector Machines[7,9,10,11]. For instance, in Morales et al.[12], 40 features obtained from finger widths for 3 finger are used to train a supported vector machine (SVM). Adán et al.[13] use a hand natural reference system in order to make the system robust against different hand poses, and the classification is based in a time averaged feature vector. This system is based on a webcam. Sánchez-Reillo et al.[14] use 25 features, such as finger widths, finger and palm heights, finger deviations and angles of the interfinger valleys with respect to the horizontal, modeling them using Gaussian mixtures. Jain et al.[15] use an imaging scheme to select 16 features, such as the length and width of the fingers, the aspect ratio of the palm to the fingers, and the thickness of the hand. Öden et al.[16] use geometric features and finger shapes. The finger shapes are modeled using fourth degree polynomials. They obtain 16 features that are compared using the Mahalanobis distance.

Hand contour based systems use the hand silhouette to perform the matching. Yoruk et al.[17] use 2048 points of contour coordinates to construct a raw feature vector and independent component analysis features are used in the identification and verification tasks. Woodard et al.[18] use shape indices based on 3D shape curvature and a match score based in correlation coefficients between shape descriptors. Finally, palm print systems use the palm silhouette lines for matching, frequently in combination with geometric measures. For instance, Khangarad et al.[19] use a 3D digitizer to extract intensity and range images from the user hand and then multimodal palm print and hand geometry features are obtained for matching. Kumar et al.[20] describe a bimodal biometric system using hand geometry and palm print information and propose an strategy to fuse both sources.

It is also important to present the evolution of the hand shape recognition systems from the operation point of view, specifically, the image acquisition system. The early hand recognition systems were constrained and contact based, using a platform and pegs or guides to situate the user hand[14,21]. Next generation were unconstrained systems, but yet with the necessity of placing the hand on a platform or scanner[13,22]. Finally, modern hand shape identification systems are unconstrained and contact free[23,24]. The hand position is free and there are no platforms or scanners to situate the user hand. Contactless hand recognition systems are increasingly receiving attention because of their better user acceptability, and their capability to be extended to daily devices such as smartphones. Finally, hand recognition using low-cost devices is an important issue. For instance, Santos-Sierra et al.[25] present an algorithm to segment hand images using multilayer graphs and Mostayed et al.[26] use low resolution hand images and compute a set of position invariant features using the Radon transform.

Our approach in this paper is to use a contact free (in-air) device to retrieve the palm parameters. We will explore the identification information contained in different geometric palm features and we will use different classifiers to show the feasibility of building an effective identification system for smart spaces applications.

3. Defining a hand shape-based identification strategy

In particular, the service scenario that we are envisioning in this work considers non-critical applications that can benefit from the availability of a straight, fast and usable identification process. Let's think on a user that arrives home, sits on the sofa and decides to keep on watching the film s/he did not finished the night before. The sofa is a smartized object which enables room control, thus just by waving his hand on the sofa arm, the room is able to recognize who the user is and, by a subsequent gesture chain, interpreting the user desire. With this idea in mind, our identification system needs to provide:

- Contactless hand shape-based identification. The identification capability will be integrated in a gesture recognition system. For this reason, the identification has to be done on in-air input.
- Accuracy enough for non-critical applications. The applications to deploy on top of the identification systems are related to interaction and personalization of smart environments. Being not critical applications, the global performance of the id system undoubtedly conditions the user experience.
- Easiness of use. The identification process has to be as simple as possible, providing enough feedback cues to the user in order for him/her to control the interaction.
- Minimum training. Supervised algorithms require the users to train them to work. Although the training stage may be unavoidable to reach sufficient accuracy, the identification method must reduce it to a minimum.
- Real-time response. The identification stage must be as quick as possible, providing immediate feedback.
- Robustness. The solution has to work coherently under different scenarios with external different conditions. For example, the environment may vary its illumination conditions.
- Scalability for smart environment-like scenarios. The solution has to be validated with a number of users that is considered reasonable for smart space applications.

Our objective is to define a solution (device, hand features and classification algorithms) that fulfills the requirements above.

3.1. Device, hand features and algorithms

The technological sensor choice to implement the identification system has been the Leap Motion sensor. Leap Motion, developed by the same named company, is a small USB peripheral device that supports hand and finger motions as input with no hand contact or touching. Launched to market in 2013, it was initially conceived to interact with a computer. Two monochromatic infrared cameras and three infrared LEDs, which generate a 3D dot pattern of IR light, compose the sensor. From the comparison of the 2D frames generated by the two cameras, dedicated software in the computer synthesizes the hand's 3D position data. The coordinate system used by this device is a Cartesian coordinate system with its origin placed in the Leap Motion's center (Fig. 1a).

The sensor can reach up to 200 frames per second in the best conditions. Every frame delivers information about the hands by comparing the IR scenes against an internal hand model. The model defines that each hand has five fingers formed by four bones (metacarpal, proximal, intermediate and distal phalanx) except for the thumb, which is formed by three (proximal, intermediate and distal phalanx). In this work, we take benefit from the API provided by Leap Motion, which can recognize hands, fingers, arms and tools (straight cylindrical objects longer and with a smaller radius fingers) over it. For each finger and bone, the API provides its width and lengths. Furthermore, it is possible to get information also from the palm and wrist width, the palm orientation or the point direction for each finger. These libraries also allow recognizing four predefined gestures (swipe, key tap, screen tap and circle), provide the images acquired by the two cameras and discern between the two hands that the user may be using. Fig. 1b and 1c shows the hand view from a visualization application using the provided features.

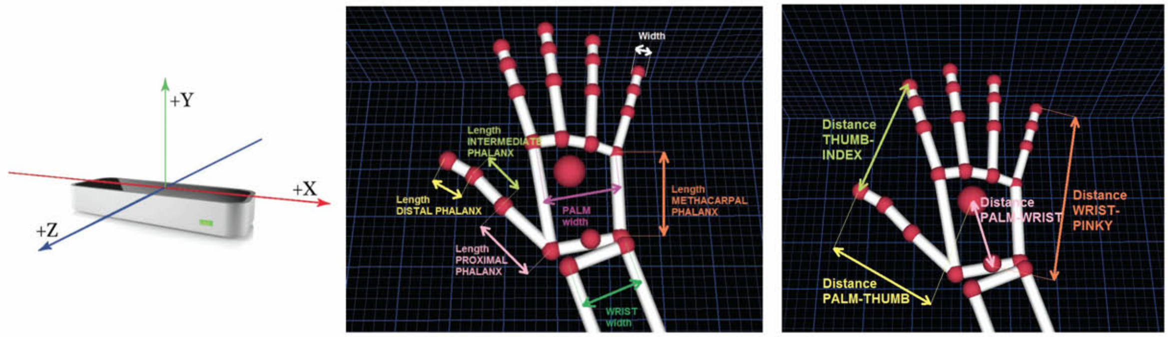

Fig. 1. (a) Leap Motion and its axis; (b) Hand length-width based features; (c) Hand distance features.

Leap Motion's API directly enables to extract sufficient geometrical hand features to attempt user identification. Taking into consideration existing literature (Section 2), the classification strategies detailed below will initially work on 52 geometric hand features: a) intrinsic morphological hand features, i.e. finger (5 features) and phalanx lengths (19) and finger, palm and wrist widths (7) (Fig. 1b) and b) pose hand features, i.e. intra-hand distances between the fingertips (10), from the palm to the fingertips (5), from the wrist to the fingertips (5) and from the wrist center to the palm center (1) (Fig. 1c). The use of intra-hand distances assumes that the user is coherent in the way s/he extends the hand over the sensor, as it relies on the relative distances between difference reference points within the hand (fingertips, palm center, wrist center).

On these features, we aim at analyzing different classification techniques to choose the most suitable one for our purposes. The supervised classification process includes a training phase in which a set of n pattern feature vectors $\{x_1, x_1, \ldots, x_n\}$ (in our case, the features described in the previous section) are gathered, assigning them to the n classes to classify (in our case, the users' identities), finally building the classification model. After taking a real-time sample, a feature vector y is computed. The vector y is in then compared to the pattern feature vectors using different strategies and finally, the class that is identified to be the 'nearest' to the feature vector y is given as the classification result. Taking the existing literature as reference point, we have decided to study the performance of well-know classification algorithms, such as Nearest Neighbor, neuronal networks (Multilayer Perceptron), Support Vector Machine (SVM), Logistic Regression and tree-based algorithms (such as functional trees or logistic trees) for our classification problem. We expect some of these methods to provide good or very good accuracy, while delivering different performance in terms of model building and classification time (computational cost).

3.2. Dataset description and test environment

In order to make our choice of the most adequate classification algorithm to implement a real-time solution for contactless identification, we have gathered a dataset of hands snapshots by using Leap Motion. 21 users -18 males and 3 females, with ages between 23 and 53- were involved in the data gathering (Nov. 2014). A hand-image recording tool was designed and implemented to enable feature recording (Fig. 2). For each user, 40 snapshots both for his/her right and left hands were recorded (80 samples x 21 users in total); in each snapshot, 52 hand features were collected (89.040 features). It took 20-25 minutes for each user to provide the 80 required samples.

The interface in Fig. 2 provides real-time feedback to help the user to place the hand in the optimal pose for recording the hand features, in terms of position and orientation. This pose, usually referred as ‘sweet spot’, makes possible to reduce the number of useless samples. The interface is configured to provide real time indications for the user to place the hand at 18-21 cm over the Leap Motion device (y axis), in the middle of it (x axis between -1.5 and 1.5 cm) and slightly advanced with respect to the device (z axis between -3 and 7 cm). This pose guarantees the best vision of the hand, as the most accurate measurements are obtained when the hand is between 15 and 25 cm and in the negative values of the z axis[27]. The indications to the user are shown through red tags that are activated in case the hand is not in the perfect position; the tags’ positions in the screen indicate the user in which direction to move the hand. The user will also receive feedback to correct the pose if the hand is not sufficiently parallel to the device. The interface includes information that is relevant for the developer (e.g. the camera views), but that will not be part of the real time system. When the user’s hand is correctly placed, the system takes a snapshot and a counter indicates that the hand has been recorded. The user has to take the hand out from the vision of the device and place it correctly again, for the next snapshot. Users rapidly get their own references to correctly place the hand, so the process is much faster as the user takes practice.

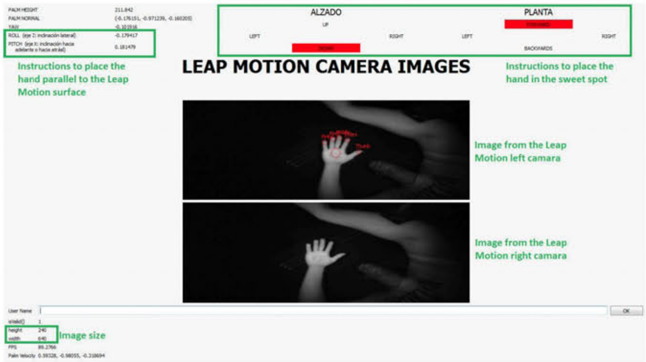

Fig. 2. Training application. In this case, the user has to move the hand down and forward to fulfill the sweet spot indications.

4. Performance analysis and comparison

Once collected, the final database has been post-processed to serve as input for the well-known Weka Open Source Data Mining Software. The use of Weka facilitates doing fast prototyping of different classification solutions, making possible to know in advance e.g. about their accuracy, time to build the model, feature relevance and training sensitivity. All algorithms that we are planning to use have a configurable Weka implementation available (Table 1).

For the experiments, Weka 3.6.11 was running in a HP Z1 Workstation (CPU 3.3GHz, RAM 8GB). Table 2 gathers the percentage of Correct Classified Instances (CCI) applying 10-fold cross validation on the right and left hand datasets (Leap Motion correctly estimates the type of hand even if turned up). In any case, we show results using the information from a) all the gathered features (52), b) those containing the distance attributes (21) or c) a subset of this latter group (11). This subset has been build from a Weka analysis on the most meaningful attributes. Three different implementations of the Best-First search algorithm (BestFirst, GreedyStepwise and LinearForwardSelection) have selected as most meaningful features 11 distances: palm-thumb, palm-pinky, wrist-pinky, thumb-index, thumb-ring, index-middle, index-ring, index-pinky, middle-ring, middle-pinky. Rank-Search adds thumb-middle to the list. And the Ranker method undoubtedly ranks all the distance features as the most meaningful for the classification. Thus distance-based features globally contain more information.

Table 1. Weka available implementations for the selected algorithms. More information available at http://www.cs.waikato.ac.nz/ml/weka/.

Classification algorithm (Weka type)	Weka available implementations
Nearest-neighbor (Rule)	NNge, Nearest-neighbor-like algorithm using non-nested generalized exemplars.
Nearest-neighbor (Lazy)	IB1. Uses Euclidean distance.
Multilayer perceptron (Func.)	Uses backpropagation to classify instances.
SMO (Functions)	John Platt's sequential minimal optimization algorithm for training a support vector machine classifier.
Logistic (Functions)	Class for building and using a multinomial logistic regression model with a ridge estimator.
Simple Logistic (Functions)	Class for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models.
FT – Functional (Trees)	Classification trees that could have logistic regression functions at the inner nodes and/or leaves.
LMT-Logistic Model (Trees)	Class. trees with logistic regression functions at the leaves.

As the reader will notice, the use of distance-based features maintains or even increases the percentage of correct classified instances. For example, in the case of IB1, the CCI increases 2.4 points in the right hand case and 2.6 points for the left hand. NNge and SMO are also positively affected (in the case of the right hand, the CCIs are improved 1.78 and 1.19 points). The rest of methods keep or vary very slightly their performance. The use of 11 features instead of 21 provides a similar accuracy.

Table 2. Correct classified instances (%) and time to build the model on 52, 21 and 11 input features. Time is provided for right hand dataset.

	52 features			21 features			11 features		
	Right hand	Left hand	Time (s)	Right hand	Left hand	Time (s)	Right hand	Left hand	Time
NNge	89,05	90,36	0,35	90,83	92,38	0,22	90,83	91,55	0,14
IB1	92,26	90,48	0,01	94,64	93,1	0,01	95,12	93,93	0,01
Multi. Perce.	96,67	95	34,91	96,79	95,71	13,22	95,60	94,26	9,12
SMO	91,07	89,05	0,56	92,26	91,55	0,53	90,71	91,31	3,24
Logistic	96,43	94,64	16,59	96,31	95,12	3,09	93,93	93,57	3,15
Simple Log.	96,43	94,76	13,69	94,43	94,88	6,77	96,07	93,81	5,05
FT	95,12	93,93	5,17	95	93,93	2,49	93,93	92,98	1,64
LMT	96,43	95	44,08	96,43	94,88	21,58	96,07	93,81	31,48

Let's consider the right hand dataset, which will be the most probable to use in real settings (most of the potential users will be right-handed, as the percentage of left-handed in the world population is between 8%-13%). In this case, the algorithms that perform better (>96%) both for 52 and 21 features are Multilayer Perceptron, Logistic and the Logistic Model Tree. Simple strategies, such as Nearest Neighbor (IB1), reach a reasonable 94% accuracy that is maintained in the case of 11 features.

With respect to necessary time to build the model, which is relevant as the model calculation has to be accomplished each time that a new user is included in the database, Multilayer Perceptron, LMT, Logistic and Logistic Model Tree are the slower strategies. The model-building time is naturally reduced with the number of features to consider in the classification process. In the case of the right hand dataset, the time to build the model is reduced up to 81% for the Logistic method, 62% for Multilayer Perceptron and around 50% for Simple Logistic and Tree-based methods when moving from 52 to 21 features. The IB1 (NN) method is the only one that remains unaffected, due to its very small needed time. Once the classification model is built, the classification result is provided immediately with the handled data volumes.

An important aspect for the real-time system is to have a reference on the minimum number of needed training samples, as this aspect directly affects the user experience (obviously users are not willing to train the systems). Fig. 3a shows how the different algorithms perform with a decreasing number of samples. The experiment has been configured to use a variable number of training samples and the same test dataset of 10 samples to check the performance for all the iterations. The 21-features right hand dataset has been used. Globally, logistic methods, multilayer perceptron and trees are more robust to the decreasing number of training samples, while NN-methods and SVM algorithms reduce their accuracy more significantly. Training with a single sample is not feasible, but there is a significant change when 2-3 samples are considered for training. If using the first group of algorithms, at least 5 trainings are needed to reach a CCI above 80%. For the second group of algorithms, 10 samples are needed to reach this same CCI. With respect to the time to build the model, the slower method is LMT (50,17 seconds for 30 training samples), followed by the multilayer perceptron (25,6 sec). The fastest are the NN methods (<1 sec).

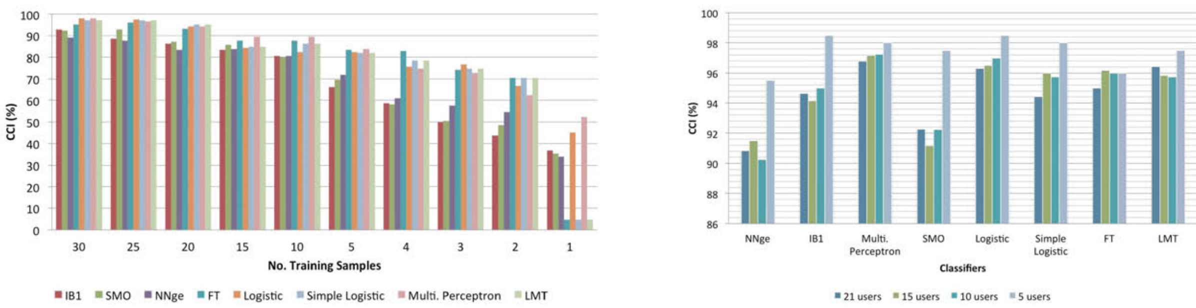

Fig. 3. Classifiers performance taking into consideration: a) the number of training samples and b) the users in the database.

Finally, it is also important to know how the different algorithms perform with respect to the number of users in the database. Scalability is key for many identification systems. In this case, our design requirements need that the identification system works well on a not too large number of users. Fig. 3b gathers some results in this direction. Experiments have been carried out on the right-hand 21 features dataset, with all the samples and 10-fold cross validation. All algorithms achieve their better performance with the smallest number of users (5), except FT that experiences a very small variation (+0.1%). Some algorithms perform more stably than others. It is the case of Multilayer Perceptron or the Trees, which CCI is around 1% lower when comparing the 5-users dataset and the 21 users dataset. On the other hand, SMO losses 5.2%, NNge 4.7%, IB1 3.9%, Logistic 2.2% and Simple Logistic 3.6% when comparing these two situations. Some other algorithms are very sensitive to the change from 5 to 10 users, then experiencing small variations on the CCI in the subsequent situations. It is the case of NN methods and SVM.

5. Conclusions and further work

In this paper, we have explored different classification algorithms that enable building a real-time in-air hand shape identification system, to be integrated with non-critical smart space applications. Taking the output of the Leap Motion API as a reference, it has been possible to compare the significance of intrinsic morphological hand features vs. pose hand features, being this second group more relevant for the classification process. The use of 'sweet spot' feedback has been crucial to obtain a reasonable number of Correctly Classified Instances for all the tested algorithms. Although the different methods obtain a variable accuracy, this aspect does not seem conclusive enough to discard any classification option, as CCIs are between 90,8% and 96,8%. The time to build a model neither seem a differentiator factor, as this process can be done out of the real-time execution. Moreover, with the reduction in the number of features, the time to build the model is significantly lowered. With respect to the need of training, it has been shown that more than 1 iterations are needed, and ideally, more than 5. An open issue is how to introduce machine-learning algorithms that may contribute to the system performance by dynamically integrating service iterations into the classification models. Results regarding the variation of the CCI depending on the number of users experiment a modification of around 5,3% and are satisfactory enough for medium-scale applications in smart environments. After ranking and giving weights (0-7) to the methods depending on their performance with respect to CCI, time to build the model, response to training and scalability on the right hand dataset, they remain as follows: FT (22 points), LMT (18), Logistic (17), MP (16), Simple Logistic (15), IB1 (13), NNge (11) and SMO (8). Note that no criteria has been prioritized, but equally considered. As the reader will notice, there are not sufficient reasons to defend a single option, but trees are offering the best global performances although may scale badly for an increasing number of users. Nearest Neighbor methods are very sensible to training, but once done, they perform reasonably. Multilayer Perceptron is an option that comes to be robust with respect to training and scales well.

Further work includes the implementation of a real-time classifier, its integration in a Leap-based gesture recognition system to interact in a smart space and the assesment of the user experience. Apart from the challenges related to the classification, issues such the implementation of the sweet spot in a non-invasive and integrated solution in the smart space are relevant (e.g. multimodal strategies may be needed when visual information is not adequate). These collateral aspects will be key to achieve an acceptable solution from the user side.

References

1. Wang X, Bernardos AM, Besada JA, Metola E, Casar JR. A Gesture-Based Method for Natural Interaction in Smart Spaces. Accepted for publication in *Journal of Ambient Intelligence and Smart Environments.*
2. Sodhi GS, Kaur J. Indian civilization and the science of fingerprinting. *Indian Journal of Traditional Knowledge* 2003;**2**:126-136.
3. Miller RP. Finger dimension comparison identification system. US Patent No. 3576538, 1971.
4. Jain AK, Flynn P, Ross AA. *Handbook of biometrics*. Springer-Verlag NY; 2007.
5. Kong A, Zhang D, Kamel M. A Survey of palm print recognition. *Pattern Recognition* 2009;**42**:1408-1418.
6. Duta N. A survey of biometric technology based on hand shape. *Pattern Recognition* 2009;**42**:2797-2806.
7. Kumar A, Zhang D. Hand-Geometry Recognition Using Entropy-Based Discretization. *IEEE Trans. on Info. Forensics& Sec* 2007;**2**:181 –187.
8. Wong RLN, Shi P. Peg-free hand geometry recognition using hierarchical geometry and shape matching. *IAPR Workshop on Machine Vision Applications,* 2002. p . 281-284.
9. Kumar A, Zhang D. Personal recognition using hand shape and texture. *IEEE Transactions on Image Processing* 2006, **15**:2454-2461.
10. Yuan Y, Barner K. An Active Shape Model Based Tactile Hand Shape Recognition with Support Vector Machines. *40th Annual Conference on Information Sciences and Systems*; 2006. p. 1611- 1616.
11. Guo JM, Liu YF, Mei-Hui C, Chia-Chu W, Thanh-Nam L. Contact-free hand geometry identification system. *18th IEEE International Conference on Image Processing (ICIP)*; 2011. p. 3185-3188.
12. Morales A, Ferrer M, Díaz F, Alonso J, Travieso C. Contact-free hand biometric system for real environm.*16th EUSIPCO Procs, 2008,* p.1-5.
13. Adán M, Adán A, Vázquez A., Torres R. Biometric verification/identification based on hands natural layout. *Image and Vision Computing* 2008, **26**:451-465.
14. Sánchez-Reillo R, Sánchez-Ávila C, González Marcos A. Biometric identification through hand geometry measurements. *IEEE Trans. Pattern Anal. Mach. Intell* 2000; **22**:1168-1171.
15. Jain AK, Ross A, Pankanti S. A prototype hand geometry based verification system. *Proc. 2nd Int. Conf. Audio&Video-Based Biometric Person Authentication* 1999; p.166-171.
16. Öden C, Erçil A, Pabhakar S. Combining implicit polynomials and geometric features for hand recognition. *Pattern Recognition Letters* 2003;**24**:2145-2152.
17. Yoruk E, Dutagaci H, Sankur B. Hand biometrics. *Image and vision computing* 2006;**24**:483-497.
18. Woodard DL, Flynn PJ. Finger surface as a biometric identifier. *Computer vision and image understanding* 2005;**100**:357-384.
19. Kanhangad, V, Kumar, A, Zhang, D. Contactless and pose invariant biometric identification using hand surface. *IEEE Trans. on Image Processing* 2011;**20**:1415-1424.
20. Kumar YA, Wong DCM, Shen HC Jain AK. Personal verification using palmprint and hand geometry biometric. *Proc. 4th Int. Conf. Audio Video-Based Biometric Person Authentication* 2003. p. 668-678.
21. Jain A, Duta N. Deformable matching of hand shapes for user verification. *Int. Conference on Image Processing ICIP* 1999;**2**:857-861.
22. Ferrer M, Fabregas J, Faundez M, Alonso J, Travieso C. Hand geometry identification system performance. *43rd Annual International Carnahan Conference on Security Technology*, 2009. p. 167-171.
23. Zheng G, Wang CJ, Boult T. Application of projective invariants in hand geometry biometrics. *IEEE Transactions on Information Forensics and Security* 2007;**2**:758-768.
24. de Santos Sierra A, Casanova J, Avila C, Vera V. Silhouette-based hand recognition on mobile devices. *43rd Annual International Carnahan Conference on Security Technology*, 2009. p. 160-166.
25. Santos-Sierra D, Arriaga-Gómez MF, Bailador G, Sánchez-Avila C. Low computational cost multilayer graph-based segmentation algorithms for hand recognition on mobile phones. *Proc. International Carnahan Conference on Security Technology (ICCST)*. 2014, p. 1-5.
26. Mostayed A, Kabir M, Khan S, Mazumder M. Biometric authentication from low resolution hand images using radon transform. *12th International Conference on Computers and Information Technology,* 2009. p. 587-592.
27. Guna J, Jakus G, Pogačnik M, Tomažič S, Sodnik J. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors* 2014;**14.2**:3702-3720.